

## CLAIMS

What is claimed is:

1. A method for improving database concurrency comprising:
  - receiving a request to access data;
  - determining a data locality within a database utilizing unique data keys;
  - determining a data locking level based on a deadlock history corresponding to the data locality;
  - providing access to the data while locking part of the database based on the data locking level.
2. The method of claim 1 wherein the database comprises a plurality of data localities.
3. The method of claim 1 wherein the deadlock history comprises a number of deadlocks in a predetermined period of time corresponding to each data locality from a plurality of data localities within the database.
4. The method of claim 1 wherein the deadlock history comprises a number of successful data accesses without a deadlock in a predetermined period

of time corresponding to each data locality from a plurality of data localities within the database.

5. The method of claim 1 wherein the data locality is a database object.
6. The method of claim 1 wherein the data locality is a database page.
7. The method of claim 1 wherein the data locality is a database file.
8. The method of claim 1 wherein the determining the data locking level comprises determining the data locking level utilizing the deadlock history and predetermined deadlock thresholds.
9. The method of claim 8 wherein the determining the data locking level further comprises comparing the deadlock history to the predetermined deadlock thresholds.
10. The method of claim 1 wherein the unique data keys comprise user identification keys.
11. The method of claim 1 wherein the unique data keys comprise mailbox identification keys.

12. The method of claim 1 wherein the unique data keys comprise email message identification keys.

13. The method of claim 1 wherein the receiving the request to access data comprises receiving the request to access data from a client.

14. The method of claim 1 wherein the database is located on a server.

15. A method for improving database concurrency comprising:

assigning a first unique key to a user of an email system located on a server;

assigning a second unique key to a mailbox of a plurality of mailboxes corresponding to the user;

assigning a third unique key to an email message of a plurality of email messages corresponding to the mailbox;

receiving a request to access data from the user;

determining a data locality of user requested data utilizing the unique identification keys;

comparing a deadlock history corresponding to the data locality to a plurality of deadlock threshold levels;

determining a data locking level based on a result of the comparing; and

locking the database based on the data locking level.

16. The method of claim 15 wherein the deadlock history comprises a number of deadlocks occurred in a predetermined period of time, the deadlocks corresponding to each data locality from a plurality of data localities within the database.
17. The method of claim 15 wherein the deadlock history comprises a number of successful data accesses without a deadlock occurred in a predetermined period of time, the successful data accesses corresponding to each data locality from a plurality of data localities within the database.
18. The method of claim 15 wherein the at least two deadlock threshold levels comprise a maximum allowed deadlocks in a predetermined period of time.
19. The method of claim 17 further comprising conservatively locking the database when deadlock history represents a number of deadlocks occurred in a predetermined period of time greater than the maximum allowed deadlocks.

20. The method of claim 15 wherein the determining the data locality comprises utilizing a hashing algorithm to determine the data locality.

21. A method for improving database concurrency comprising:

assigning a first unique key to a user of a data system located on a server;

assigning a second unique key to a mailbox of a plurality of mailboxes corresponding to the user;

assigning a third unique key to an email message of a plurality of email messages corresponding to the mailbox;

receiving a request to access data from the user;

determining a data locality of user requested data utilizing the unique identification keys;

comparing a deadlock history corresponding to the data locality to a plurality of deadlock threshold levels;

determining a data locking level based on a result of the comparing; and

locking the database based on the data locking level.

22. The method of claim 21 wherein the data system is an email system.

23. The method of claim 21 wherein the deadlock history comprises a number of deadlocks occurred in a predetermined period of time, the deadlocks corresponding to each data locality from a plurality of data localities within the database.
24. The method of claim 21 wherein the deadlock history comprises a number of successful data accesses without a deadlock occurred in a predetermined period of time, the successful data accesses corresponding to each data locality from a plurality of data localities within the database.
25. The method of claim 21 wherein the at least two deadlock threshold levels comprise a maximum allowed deadlocks in a predetermined period of time.
26. The method of claim 23 further comprising conservatively locking the database when deadlock history represents a number of deadlocks occurred in a predetermined period of time greater than the maximum allowed deadlocks.
27. The method of claim 21 wherein the determining the data locality comprises utilizing a hashing algorithm to determine the data locality.

28. An apparatus for improving database concurrency comprising:
  - a data locator to determine a user-requested data locality within a database utilizing unique data keys;
  - a deadlock analysis module to determine a data locking level by comparing a number of deadlocks corresponding to the data locality occurred in a predetermined period of time to a deadlock threshold level and to lock the database based the data locking level.
29. The apparatus of claim 28 wherein the database comprises a plurality of data localities.
30. The apparatus of claim 28 wherein the deadlock threshold level represents a maximum allowed deadlocks within the predetermined period of time.
31. The apparatus of claim 28 wherein the data locality is a database object.
32. The apparatus of claim 28 wherein the data locality is a database page.
33. The apparatus of claim 28 wherein the data locality is a database file.
34. The apparatus of claim 28 wherein the deadlock analysis module further configured to conservatively lock the database if the number of deadlocks

occurred in a predetermined period of time is greater than the deadlock threshold level.

35. An apparatus for improving database concurrency comprising:

means for receiving a request to access data;

means for determining a data locality within a database utilizing unique data keys;

means for determining a data locking level based on a deadlock history corresponding to the data locality;

means for providing access to the data while locking part of the database based on the data locking level.

36. The apparatus of claim 35 wherein the database comprises a plurality of data localities.

37. The apparatus of claim 35 wherein the deadlock history comprises a number of deadlocks in a predetermined period of time corresponding to each data locality from a plurality of data localities within the database.

38. The apparatus of claim 35 wherein the deadlock history comprises a number of successful data accesses without a deadlock in a predetermined



period of time corresponding to each data locality from a plurality of data localities within the database.

39. The apparatus of claim 35 wherein the data locality is a database object.

40. The apparatus of claim 35 wherein the data locality is a database page.

41. The apparatus of claim 35 wherein the data locality is a database file.

42. The apparatus of claim 35 wherein the means for determining the data locking level comprise means for determining the data locking level utilizing the deadlock history and predetermined deadlock thresholds.

43. The apparatus of claim 42 wherein the means for determining the data locking level further comprise the means for comparing the deadlock history to the predetermined deadlock thresholds.

44. A processing system comprising:

a processor; and

a storage medium having stored therein instructions which, when executed by the processor, cause the processing system to perform a method comprising:

receiving a request to access data;  
determining a data locality within a database utilizing  
unique data keys;  
determining a data locking level based on a deadlock history  
corresponding to the data locality;  
providing access to the data while locking part of the  
database based on the data locking level.

45. The method of claim 44 wherein the database comprises a plurality of data  
localities.

46. The method of claim 44 wherein the deadlock history comprises a number  
of deadlocks in a predetermined period of time corresponding to each  
data locality from a plurality of data localities within the database.

47. The method of claim 44 wherein the deadlock history comprises a number  
of successful data accesses without a deadlock in a predetermined period  
of time corresponding to each data locality from a plurality of data  
localities within the database.

48. The method of claim 44 wherein the data locality is a database object.

49. The method of claim 44 wherein the determining the data locking level comprises determining the data locking level utilizing the deadlock history and predetermined deadlock thresholds.

50. The method of claim 49 wherein the determining the data locking level further comprises comparing the deadlock history to the predetermined deadlock thresholds.

51. The method of claim 44 wherein the unique data keys comprise user identification keys.